Notes 6: Advanced Linux Commands

awk

Definition:

A versatile programming language for working on files.

Usage

`awk [options] 'script' [file...]`

Examples:

- Print the first column of a file:
  - `awk '{print $1}' file.txt`
- Sum and print the values of the first column:
  - `awk '{sum += $1} END {print sum}' file.txt`
- Filter and print lines where the first column is greater than 100:
  - `awk '$1 > 100' file.txt`

sed

Definition:

Stream editor for filtering and transforming text.

Usage

`sed [options] 'script' [file...]`

Examples:

- Replace "day" with "night" in a file:
  - `sed 's/day/night/' file.txt`
- Delete lines containing "error":
  - `sed '/error/d' file.txt`

less

Definition:

A program similar to more, but which allows backward movement in the file as well as forward movement.

Usage

`less [options] [file]`

Examples:

- Open a file for interactive viewing:
  - `less file.txt`

alias

Definition:

Creates an alias for a command.

Usage

`alias name='command'`

Examples:

- Create an alias `ll` for `ls -l`:
  - `alias ll='ls -l'`
- Remove an alias:
  - `unalias ll`

>

Definition:

Redirects output to a file, overwriting the file.

Usage

`command > file`

Examples:

- Redirect the output of `ls` to a file:
  - `ls > list.txt`

>>

Definition:

Redirects output to a file, appending to the file.

Usage

`command >> file`

Examples:

- Append the output of `ls` to a file:
  - `ls >> list.txt`

|

Definition:

Passes the output of one command to another command as input.

Usage

`command1 | command2`

Examples:

- List files and filter with grep:
  - `ls | grep "txt"`
- Sort and count unique lines:
  - `cat file.txt | sort | uniq -c`